Lloyd
Wise

Patents Designs Trade Marks
London
Hong Kong
Singapore
Manchester
Beijing

Lloyd Wise
200 Cantonment Road
#14-01 Southpoint
Singapore 089763

Telephone:	(+65) 6227 8986
Facsimile:	(+65) 6227 3898
E-mail:	mail@lloydwise.com.sg
Website:	http://www.lloydwise.com
Registration No:	49967800A

Austrian Patent Office
Dresdner Straβe 87
A-1200 Vienna
Austria

Our Ref: FP1897/ES

Your Ref: PCT2003000235

27 February 2006

FAX COPY - CONFIRMATION BY COURIER

No. of Pages 27 **Your Fax No. 001 43 1 534 24200** **Our Fax No. (65) 6227 3898**

Dear Sir,

Re: PCT Patent Application
entitled "A METHOD OF UPDATING A DATA SOURCE FROM TRANSFORMED DATA"
in the name Vibrasoft Pte. Ltd.

Thank you for your letter of 26 January 2006 enclosing the Written Opinion, in response to which we enclose a set of claims amended under Article 34. Claims 1 to 4, 8 to 10, 14, 17 to 21, and 28 to 40 have now been deleted and the remaining claims narrowed in scope and re-numbered.

Amendments

Original claim 5 has been combined with original claim 6 and further limited, and is now amended main claim 1. Basis for the new limitations is found in page 14, 35 and Figure 4.

In particular, the feature of 'unique' has been removed from the 'identifier' in claim 1, which is supported in the embodiment of Figure 6, showing how two nodes having the same identifier but coming from two different source documents may be distinguished based on the root document identifiers.

Original claim 7 has been further limited and is now amended claim 2.

Original claim 11 is now amended claim 3, the amendments made being for clarification and cosmetic purposes. Thus, claim 3 finds basis in original claim 11.

Similarly, original claims 12 and 13 are now amended claims 4 and 5 respectively, the amendments made being for clarification and cosmetic purposes.

New claim 6 finds basis in original claim 38.

New claim 7 finds basis in original claim 1.

Original claim 15 is now amended claim 8 and has been further limited by a feature found in page 14 and Figure 4.

Original claim 16 is now amended claim 9 and has been amended for clarification and cosmetic purposes.

New claim 10 finds basis in original claim 39.

New claim 11 finds basis in original claim 1.

Original claims 22 and 23 have been combined as amended claim 12, the scope of which finds basis in page 14 and Figure 4.

Original claim 24 is now amended claim 13 and has been amended for clarity, the basis for which is found in page 35.

Original claims 25 to 27 have also been amended for clarification and cosmetic purposes and are now amended claims 14 to 16.

New claim 17 finds basis in page 36 line 20 to 21.

New claim 18 finds basis in Figure 5.

New claim 19 finds basis in original claim 1.

New claim 20 claiming that the data source is an XML document finds basis throughout the description.

A copy of the claims marked-up as amended is enclosed for your reference, as well as a set of amended claims in neat.

The Written Opinion

The examiner objects to all the claims on file on the basis that the present invention is not novel in the light of US 2003149935 (**D1**).

The examiner also considered the MATE Workbench document (**D2**), downloaded from http://citeseer.ist.psu.edu/290011.html and an email thread in http://www.stylusstudio.com/xmldev/200007/post90030.html (**D3**).

The examiner also refers to US 2004044965 (**D4**) although he notes that D4 is of category 'X,E' in the international search report, i.e. D4 is filed before but published after the filing date of the present application. Thus, D4 is not citable for international examination according to PCT rule 64.

The present invention, as claimed in the amended claims, provides the possibility of editing an XML source document, including updating the document content and modifying the document structure (page 13 line 1: "text nodes, attribute nodes or element nodes"). By modification of document structure, it means nodes are deleted or added to the XML source document. The modifications are initiated by actions on a transformed version of the document, which is typically but not necessarily a HTML document displayed in a web browser. For example, clicking on a specific piece of information displayed in the browser is translated into an operation on a specific node in the XML source document.

To link the information in the HTML document to the nodes in the XML source document, the invention provides a set of unique node identifiers to annotate the XML source document. The identifiers are generated and inserted into the nodes in the XML source document when the XML source document is used to create a HTML document for the first time. Thus, the identifiers are inherited by the nodes in the HTML document (page 11 line 23 to page 12 line 7).

The identifiers are generated once for the nodes in the XML source document and are re-used consistently for the same nodes, i.e. the identifiers are persistent. Only nodes newly created in the XML document are given new identifiers while 'old' nodes retain the identifiers once given. Therefore, the identifiers of the existing nodes are not affected by any change in the structure of the XML document (page 13 lines 14 to 18). It follows that the identifiers are not indicators of relative node positions in the XML source document, as is the case in some prior art.

Accordingly, the present invention now claims:

1. A method of modifying a node-based data source comprising the steps of:
associating nodes in the node-based data source with a set of first identifiers;
selecting one of the nodes in the node-based data source by selecting a corresponding node from a separate and transformed version of the data source, the corresponding node in the separate and transformed version having the same first identifier as the selected node in the node-based data source;
deleting the selected node in the node-based data source by reference to the selected node's identifier or inserting a new node into the node-based source in a position relative to the selected node's position;
wherein
the first identifiers are used to identify the same nodes in the node-based data source after the deletion of the selected node or insertion of the new node in the node-based data source.

8. A data source structured to operate as a node-based data source having at least one node associated with an identifier; wherein
the same identifier is used to identify the same node when other nodes are added to or deleted from the data source.

12. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source; wherein
the identifier is used to identify the same node in the data source when other nodes are added to or deleted from the data source.

Novelty

D1 teaches how to "prepare a document required to be displayed in a browser by using a browser" (0016).. That is, users are able to update XML data in a server through action on a web browser displaying a corresponding HTML document (0054). The HTML document contains data extracted from nodes in the XML source document (0101) and the data is tagged with identification numbers (0103, 0115). A matching list is used to match the identification numbers in the HTML document to corresponding nodes in the XML source document (0104).

However, the identification numbers are inserted into the generated HTML document and not into the original XML source document; the XML source document in D1 remains un-annotated. Therefore, as the present invention has a feature of an annotated source document, the present invention is novel over D1.

Furthemore, D1 does not explain how is an identification number matched to a specific node in the XML source document, i.e. what is used to identify or specify the node in the XML source document so that the matching list may match the identification number to the node? In other words, assuming that the matching list has two columns and one column contains identification numbers, what would then be in the second column? If the second column contains an indication of the node's relative position, a change in the XML document structure would require the indication to be re-generated and the indication is thus not a persistent node identifier.

Furthermore, while D1 teaches that the identification numbers are attached anew to re-generated HTML documents (0115, 0122, 0130, 0153), D1 is silent on whether exactly the same identification number is re-used in re-generated HTML documents.

Therefore, there is no disclosure of using persistent identifiers to annotate a source document in D1. The invention as claimed in claims 1, 8 and 12, as well as the claims dependent on them, are novel over D1.

D2 describes a commercial product named the MATE Workbench, which is an annotation tool for XML codes. D2 page 16 casually mentions the need for back-pointers to link a display object to the XML source document. However, that is where the discussion stops; there is no further discussion on how the back-pointers should function, how the back-pointers may be used to effect a change in the XML document or the syntax of the back-pointers. In particular, there is no mention of annotating the XML source document with persistent identifiers. In contrast, the present invention has a feature of a source document annotated with persistent identifiers.

Therefore, claims 1, 8 and 12, as well as the claims dependent on them, are also novel over D2.

D3 is an email from a discussion thread on the topic of XML, in which the writer discussed on how to update an XML using reverse transformation initiated from an XHTML page. D3 does not disclose annotating the XML source document with identifiers. Furthermore, there is no discussion on persistent node identifiers in the XML source document. It is significant that the writer mentioned that 'no tree manipulation is allow' because if persistent identifiers are used as in the present invention, there would be no need to prevent changes to the structure of the XML document.

Therefore, claims 1, 8 and 12, as well as the claims dependent on them, are also novel over D3.

D4 was published after the filing date of the present application and is therefore not relevant prior art according to PCT rule 64, which only requires consideration of prior art published before the filing date of the application.

Nevertheless, even if D4 is relevant for the purpose of novelty, the present invention as claimed in the amended claims is also novel over D4. The reason is that the identifiers in D4 are used to link a document to corresponding source nodes in an original XML by the nodes' relative positions (see para. 0109 and Figure 19). For example, the first child node of the second child node of the first node is annotated "1.2.1". Such relative node positions change whenever there is a change in the XML document structure. Thus, the identifiers in D4 are not persistent.

Accordingly claims 1, 8 and 12, as well as the claims dependent on them, are novel over D4 even if D4 is relevant as prior art, which is not admitted here.

Accordingly, all the claims in the amended set of claims are novel over all the citations.

Inventive step

As mentioned above, D4 is not citable as prior art for international examination.

As also mentioned above, D1 only teaches using identification numbers to annotate the generated HTML document. Similarly, D2 mentions back-pointers in the generated HTML document (see page 16) and D3 mentions back-pointers in 'the resultant tree'. Thus, none of D1 to D3 mentions annotation of a source document.

Furthermore, D2 and D3 contain only fleeting suggestions on back-pointers or relative node position identifiers but not on persistent node identifiers. Thus, even if D1 is read together with D2 and D3, the skilled man will find no teaching to annotate a source document with persistent identifiers. Accordingly, there is no inspiration, motivation or suggestion in D1 to D3 for the skilled man to annotate a source document with persistent node identifiers.

Conversely, if the skilled man achieves the present invention on reading D1 to D3, the skilled man must have exercised an inventive creativity beyond D1 to D3, as he would have proposed 1) annotating the XML source document and 2) with persistent identifiers.

Accordingly, claims 1, 8 and 12 are inventive over all of D1 to D3.

Furthermore, the present invention provides the possibility of having some advantages not offered by the citations. D1 is the most detailed of the citations and probably the closest prior art. However, D1 does not sufficiently describe how may a plurality of users modify the same piece of XML data at once, i.e. when two users attempt to access the same XML node from two different HTML documents, is there a need for one or two matching lists? If two matching lists are produced for two HTML documents, then is there a need to synchronise the matching lists? As D1 is silent on this aspect, D1 does not truly provide the advantage of true multi-user access.

In contrast, the present invention provides the possibility of true multiple-user access, even to the same XML source node. Each of the users may modify the same XML node by means of the node's persistent identifier. In other words, the persistent identifiers of the present invention are absolute identifiers. Even when two users try to amend the same node at the same time, the system running the invention will not be confused as in the case in which relative node positions are used as 'back-pointers'.

Furthermore, D1's embodiments are limited to a HTML document produced by a single transformation from the XML document. Thus, it is not clear whether a HTML document produced by multiple-transformations of the XML document would require a plurality of matching lists. If it does, any change made to the HTML document will have to be tediously traced through the plurality of matching lists to reach the original source node. In contrast, the persistent node identifiers of the present invention allow a HTML document produced by multiple-transformations of the XML source document to retain a direct link to the source nodes. Thus, in comparison, the present invention possibly provides a more direct and elegant alternative to D1 and is also thereby advantageous over D1.

As D2 and D3 only contain fleeting suggestions to back-pointers, even when reading all of D1 to D3 together, there is no disclosure of how to provide true multi-user access and there is also no disclosure on how to link data in a multi-transformed document to the nodes in the source document.

Further comments

Furthermore, the applicant wishes highlight that the invention as claimed in new claim 18, i.e. composite identifiers made up of source document identifiers and node identifiers of (see embodiment of Figure 6), is also particularly advantageous over D1 to D3. The composite identifiers provide a means of tracking data in a single HTML to a plurality of XML source documents, i.e. multiple XML source documents may be amended through a single HTML page. Such a feature is not available in D1 to D3.

The examiner is mistaken to comment that Figure 5 of the present application discloses a reverse transformation. Reverse transformation means that a new XML document is generated 'backwardly' to replace an original XML document. Instead, Figure 5 illustrates using a neighbouring node to position a new node. Thus, there is no reverse transformation in Figure 5.

The examiner also points out that the method as claimed in the original claims was not useable to perform some types of transformations. However, the scope of the present invention is not in document transformations. Instead, the present invention provides the possibility of facilitating source document amendments by the persistent identifiers annotating the source document.

In summary, we submit that the present invention as claimed in the amended claims is new and inventive over D1 to D3.

Should, however, the examiner have any further concerns, we would be grateful for the issuance of a further official action to allow applicants an opportunity to address them.

Yours faithfully

Jim Greene-Kelly
Lloyd Wise, Singapore
Encls.
FP189761.doc

Claims

~~1. A method of identifying data in a node-based data source comprising the steps of annotating at least one node with a unique identifier.~~

~~2. A method according to claim 1 wherein only selected nodes are annotated with unique identifiers.~~

~~3. A method as claimed in claim 1or 2 wherein the unique identifier comprises of system names or addresses.~~

~~4. A method as claimed in claim 1or 2 wherein the unique identifier does not include system names or addresses; whereby the data source is not restricted to reside in any particular system.~~

~~5~~1. A method of modifying a node-based data source comprising the steps of:

associating ~~selected~~ nodes in the node-based data source with a set of ~~unique~~ first identifiers;

selecting one of the nodes in the node-based data source by selecting a corresponding node from a separate and transformed version of the data source, the corresponding node in the separate and transformed version having the same first identifier as the selected node in the node-based data source;

~~identifying~~ deleting the selected~~a~~ node in the node-based data source by reference to the selected node's identifier ~~to be modified~~ or inserting a new node into the node-based source in a position relative to the selected node's position; ~~by reference to its identifier; and~~

~~modifying the node data.~~

~~6. A method as claimed in claim 5~~ wherein

~~the node is identified by selecting a corresponding node from a separate transformed version of the source; wherein~~

~~the corresponding node in the separate transformed version and the node in the source document have the same identifier~~

the first identifiers are used to identify the same nodes in the node-based data source after the deletion of the selected node or insertion of the new node in the node-based data source.~~.~~

~~7~~2. A method as claimed in claim ~~5~~ 1 wherein ~~the node is identified by selecting a corresponding node from a separate transformed version of the source; wherein~~

the corresponding node in the separate and transformed version and the node in the node-based source ~~document~~ are mapped to each other by a series of further identifiers; and

each further identifier in the series representing a transformation in a series of multiple-transformation to transform the node-based source into the transformed version.

~~8. A method as claimed in claim 7 whereby the series of identifiers provide means for a proxy or firewall interface whereby the nodes in the data source is protected from unauthorised direct access.~~

~~9. A method as claimed in claims 6 or 7 wherein the separate transformed version of the data source is displayed in a desired presentation format.~~

~~10. A method as claimed in claims 5, 6 or 7 wherein the modification comprises the steps of:~~

~~deleting the selected node in the data source.~~

~~11~~3. A method as claimed in claims ~~6~~ 1 or ~~7~~ 2 wherein inserting a new node into the node-based data source ~~the modification further~~ comprises the steps of:

initiating an insertion of a ~~ing a~~ node via ~~in~~ the transformed version~~of the source~~;

updating the node-based data source by creating ~~a~~ the new ~~counterpart~~ node in the node-based data source~~corresponding to the inserted node~~;

wherein

the ~~counterpart~~ new node is positioned in the node-based data source in a position relative to the position of the identified node~~dependence upon at least one property of one or more nodes in the transformed version~~.

~~12~~4. A method as claimed in anyone of the preceding claims ~~5, 6, 7, 8, 9, 10 or 11~~ wherein the node-based data source is a marked-up language document.

~~13~~5. A method as claimed in anyone of the preceding claims ~~6, 7, 8, 9 or 11~~ wherein the transformed version ~~data source~~ is a marked-up language document.

~~14. A method as claimed in claim 6 or 7 wherein at least one node in the transformed data source does not have an identifier; whereby modification cannot be effected on the node without an identifier.~~

6. A method as claimed in any one of claims 1 to 5 wherein the data source is a transformation script.

7. A method as claimed in any one of claims 1 to 6 wherein identifiers are unique in the node-based data source.

~~15~~8. A data source structured to operate as a node-based data source having ~~wherein~~ at least one node ~~is~~ associated with an ~~unique~~ identifier~~.~~; wherein the same identifier is used to identify the same node when other nodes are added to or deleted from the data source.

~~16~~9. A data source as claimed in claim ~~12 and 15~~8 wherein the node-based data source is an XML document.

~~17. A data source as claimed in claim 13 and 14 wherein the transformed version of the data source is a marked up language document.~~

~~18. A data source as claimed in claim 18 wherein the mark up language is XML.~~

10. A data source as claimed in any one of claims 8 to 9 wherein the node-based data source is a transformation script.

11. A data source as claimed in any one of claims 8 to 10 wherein the identifier is unique in the node-based data source.

~~19. A method of annotating a transformed version of a data source comprising the steps of:~~

~~copying identifiers in the nodes in a data source to corresponding nodes in the transformed version of the data source.~~

~~20. A method as claimed in claim 19 wherein only selected identifiers are transferred to the transformed version of the data source.~~

~~21. A method as claimed in claim 20 wherein the selection criteria for the identifiers are in transformation or query scripts from which the transformed version of the data is derived.~~

~~22~~12. An identifier for ~~which is capable of uniquely~~ identifying a node in ~~the~~ a data source and also a corresponding node in a transformed version of the data source; ~~whereby~~

~~the node in the data source is mapped to the corresponding node in the transformed version of the data source.~~

~~23. An identifier as claimed in claim 22 which is:~~ wherein

the identifier is used to identify the same node in the data source when other nodes are added to or deleted from the data source.~~unchanged despite modifications to the data source structure or the data source itself; and~~

~~unchanged despite modifications to the structure of or the data in the transformed version of the data source itself.~~

~~24~~13. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in claim ~~22~~ 12 wherein

the identifier ~~further~~ is tagged with information on ~~provides a means of~~ the state of display of the transformed document; whereby the identifier is a means of serialising ~~a~~ the state of display ~~in~~ of the transformed version of the data source~~; whereby~~ such that a second transformed version of the data source has the

~~the display state of the transformed data source before an update action is returned to that~~ same state of ~~in the~~ display ~~after an update action and a refresh of the display.~~.

~~25~~14. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in claim~~s 22, 23 or 24~~ 12 or 13 wherein the ~~comprising:~~ identifier is composed of

one or more identifiers derived from user ~~names~~identity.

~~26~~15. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in anyone of claims 12 to 14~~s 22, 23 or 24~~ wherein

the identifier is composed of ~~comprising:~~ one or more identifiers derived from ~~system~~ hardware ~~names~~identity.

~~27~~16. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in anyone of claims 12 to 15 ~~s 22, 23 or 24~~ wherein the identifier is composed of ~~comprising:~~

identifiers derived from one or more identifiers derived from hardware identity ~~system names~~ and one or more identifiers derived from user identity~~identifiers derived from user names~~.

17. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in any one of claims 12 to 16 wherein

the data source is a transformation script.

18. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in any one of claims 12 to 17 wherein

the identifier comprises a portion indicating the data source;

a portion indicating the node; and

a portion indicating the parent node of the node; whereby

nodes of different data sources having the same node identifier and integrated in the transformed version are differentiable by the portions indicating the parent nodes or the data source.

19. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in any one of claims 12 to 18 wherein

identifier is unique in the data source.

20. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in any one of claims 12 to 19 wherein

the data source is an XML document.

~~28. An identifier as claimed in anyone of claims 22 to 27 which is removed from the node it identified after an update session is over.~~

~~29. A data transformation engine comprising:~~

~~means of copying identifiers of nodes in the data source and inserting the identifiers into the nodes of the transformed version of the data source; whereby~~

~~the nodes in the transformed version of the data source are mapped to their corresponding nodes in the data source.~~

~~30. A thin client system implementing the methods as claimed in claim 4 or 5, comprising:~~

~~a separate server system in which the transformation takes place;~~

~~a separate client system being a system for display of the transformed version of the data source.~~

~~31. A fat client system implementing the methods as claimed in claim 4 or 5, comprising:~~

~~a separate server system wherein the data source resides;~~

~~the separate client system which receives the data from the server system and in which the transformation of the data source takes place.~~

~~32. An industrial standard of node-based document modification adopting the methods as claimed in any one of claims 1 to 14, or any one of claims 17 to 21.~~

~~33. An industrial standard of identification of nodes in a node-based document adopting the identifiers as claimed in anyone of claims 22 to 28.~~

~~34. An industrial standard of node-based data transformation adopting the engine as claimed in claim 29.~~

~~35. A method as claimed in any one of claims 1 to 14, or any one of claims 19 to 21 wherein the data source is itself a transformation script.~~

~~36. A data source as claimed in any one of claims 15 to 18 wherein the data source is itself a transformation script.~~

~~37. An identifier as claimed in any one of claims 22 to 28 wherein the data source is itself a transformation script.~~

~~38. An data transformation engine as claimed in claim 29 wherein the data source is itself a transformation script.~~

~~39 A thin-client system as claimed in claim 30 wherein the data source is itself a transformation script.~~

~~40 A fat-client system as claimed in claim 31 wherein the data source is itself a transformation script.~~

Claims

1. A method of modifying a node-based data source comprising the steps of:

associating nodes in the node-based data source with a set of first identifiers;

selecting one of the nodes in the node-based data source by selecting a corresponding node from a separate and transformed version of the data source, the corresponding node in the separate and transformed version having the same first identifier as the selected node in the node-based data source;

deleting the selected node in the node-based data source by reference to the selected node's identifier or inserting a new node into the node-based source in a position relative to the selected node's position; wherein

the first identifiers are used to identify the same nodes in the node-based data source after the deletion of the selected node or insertion of the new node in the node-based data source.

2. A method as claimed in claim 1 wherein

the corresponding node in the separate and transformed version and the node in the node-based source are mapped to each other by a series of further identifiers; and

each further identifier in the series representing a transformation in a series of multiple-transformation to transform the node-based source into the transformed version.

3. A method as claimed in claims 1 or 2 wherein inserting a new node into the node-based data source comprises the steps of:

initiating an insertion of a node via the transformed version;

updating the node-based data source by creating the new node in the node-based data source; wherein

the new node is positioned in the node-based data source in a position relative to the position of the identified node.

4. A method as claimed in anyone of the preceding claims wherein the node-based data source is a marked-up language document.

5. A method as claimed in anyone of the preceding claims wherein the transformed version is a marked-up language document.

6. A method as claimed in any one of claims 1 to 5 wherein the data source is a transformation script.

7. A method as claimed in any one of claims 1 to 6 wherein identifiers are unique in the node-based data source.

8. A data source structured to operate as a node-based data source having at least one node associated with an identifier; wherein

the same identifier is used to identify the same node when other nodes are added to or deleted from the data source.

9. A data source as claimed in claim 8 wherein the node-based data source is an XML document.

10. A data source as claimed in any one of claims 8 to 9 wherein the node-based data source is a transformation script.

11. A data source as claimed in any one of claims 8 to 10 wherein the identifier is unique in the node-based data source.

12. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source; wherein

the identifier is used to identify the same node in the data source when other nodes are added to or deleted from the data source.

13. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in claim 12 wherein

the identifier is tagged with information on the state of display of the transformed document; whereby the identifier is a means of serialising the state of display of the transformed version of the data source such that a second transformed version of the data source has the same state of display .

14. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in claim 12 or 13 wherein the identifier is composed of

one or more identifiers derived from user identity.

15. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in anyone of claims 12 to 14 wherein

the identifier is composed of one or more identifiers derived from hardware identity.

16. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in anyone of claims 12 to 15 wherein the identifier is composed of

identifiers derived from one or more identifiers derived from hardware identity and one or more identifiers derived from user identity.

17. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in any one of claims 12 to 16 wherein

the data source is a transformation script.

18. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in any one of claims 12 to 17 wherein

the identifier comprises a portion indicating the data source;

a portion indicating the node; and

a portion indicating the parent node of the node; whereby

nodes of different data sources having the same node identifier and integrated in the transformed version are differentiable by the portions indicating the parent nodes or the data source.

19. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in any one of claims 12 to 18 wherein

identifier is unique in the data source.

20. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in any one of claims 12 to 19 wherein

the data source is an XML document.

Claims

1. A method of modifying a node-based data source comprising the steps of:

associating nodes in the node-based data source with a set of first identifiers;

selecting one of the nodes in the node-based data source by selecting a corresponding node from a separate and transformed version of the data source, the corresponding node in the separate and transformed version having the same first identifier as the selected node in the node-based data source;

deleting the selected node in the node-based data source by reference to the selected node's identifier or inserting a new node into the node-based source in a position relative to the selected node's position; wherein

the first identifiers are used to identify the same nodes in the node-based data source after the deletion of the selected node or insertion of the new node in the node-based data source.

2. A method as claimed in claim 1 wherein

the corresponding node in the separate and transformed version and the node in the node-based source are mapped to each other by a series of further identifiers; and

each further identifier in the series representing a transformation in a series of multiple-transformation to transform the node-based source into the transformed version.

3. A method as claimed in claims 1 or 2 wherein inserting a new node into the node-based data source comprises the steps of:

initiating an insertion of a node via the transformed version;

updating the node-based data source by creating the new node in the node-based data source; wherein

the new node is positioned in the node-based data source in a position relative to the position of the identified node.

4. A method as claimed in anyone of the preceding claims wherein the node-based data source is a marked-up language document.

5. A method as claimed in anyone of the preceding claims wherein the transformed version is a marked-up language document.

6. A method as claimed in any one of claims 1 to 5 wherein the data source is a transformation script.

7. A method as claimed in any one of claims 1 to 6 wherein identifiers are unique in the node-based data source.

8. A data source structured to operate as a node-based data source having at least one node associated with an identifier; wherein

the same identifier is used to identify the same node when other nodes are added to or deleted from the data source.

9. A data source as claimed in claim 8 wherein the node-based data source is an XML document.

10. A data source as claimed in any one of claims 8 to 9 wherein the node-based data source is a transformation script.

11. A data source as claimed in any one of claims 8 to 10 wherein the identifier is unique in the node-based data source.

12. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source; wherein

the identifier is used to identify the same node in the data source when other nodes are added to or deleted from the data source.

13. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in claim 12 wherein

the identifier is tagged with information on the state of display of the transformed document; whereby the identifier is a means of serialising the state of display of the transformed version of the data source such that a second transformed version of the data source has the same state of display .

14. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in claim 12 or 13 wherein the identifier is composed of

one or more identifiers derived from user identity.

15. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in anyone of claims 12 to 14 wherein

the identifier is composed of one or more identifiers derived from hardware identity.

16. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in anyone of claims 12 to 15 wherein the identifier is composed of

identifiers derived from one or more identifiers derived from hardware identity and one or more identifiers derived from user identity.

17. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in any one of claims 12 to 16 wherein

the data source is a transformation script.

18. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in any one of claims 12 to 17 wherein

the identifier comprises a portion indicating the data source;

a portion indicating the node; and

a portion indicating the parent node of the node; whereby

nodes of different data sources having the same node identifier and integrated in the transformed version are differentiable by the portions indicating the parent nodes or the data source.

19. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in any one of claims 12 to 18 wherein

identifier is unique in the data source.

20. An identifier for identifying a node in a data source and also a corresponding node in a transformed version of the data source as claimed in any one of claims 12 to 19 wherein

the data source is an XML document.